Public dCache Upgrade plan

Timur Perelmutov

Version 03

Table of Contents

Introduction

Current version of the dCache Software deployed in production on Public dCache is 1.9.2-9. This version of dCache has been deprecated. Currently recommended version is dCache 1.9.5-16. There is a number of bugs that were resolved between these two releases, and a number of performance and monitoring improvements that are present in the new release. One bug that draws attention that affects 1.9.2-9 is the possibility that two dcap clients swap the data streams, if the client start transfers of the two files from the same pool node within the same millisecond according to the pool timer. The changes between 1.9.2 and 1.9.5 versions of dCache can be found in the release notes on download pages of dCache at www.dcache.org.

In addition to the update of the core dCache software, we also plan to install new versions of the encp client from rpm and new updated versions of the real-encp.sh scripts that can work with such clients.

There was also a new pnfs rpm officially published on dCache.org. We propose to upgrade pnfs rpm at the same time as dCache. No database upgrade with dump and restore is needed.

The current target date for the upgrade of the Public dCache is March 4, 2010. The date depends on the successful completion of the Preparation, Testing and Documentation phases of the plan.

A brief description of the Fermilab dCache installation

Dcache-server rpm installs dcache jars, configuration file templates and scripts for startup, status query and shutdown in /opt/d-cache directory of a server. A number of configuration files need to be created from templates or modified in order to customize a system to run desired services on a node. Dcache data nodes a.k.a. pools require creation of the directory structure where data and control files would reside. Dcache configuration process is described in dCache book at http://www.dcache.org/manuals/Book/. This process does not scale well for a system that consist of more than a small number of nodes. In order to automate the dCache installation and configuration and to guarantee the consistent results when installing dCache service on the same nodes, dcache configuration rpms was developed by Fermila dCache team (originally by Jon Bakken) .

There are two types of fermilab dcache configuration rpms, one is for installation on the admin nodes of dcache, other is for dCache pool nodes. Their names for public dcache are dcache-pool-stken and dcache-admin-stken. Dcache configuration rpm installs a small number of configuration files and scripts into /opt/d-cache/admin area, and runs dcache-pool-install or dcache-admin-install scrip depending on its type.

The logic of configuration scripts dependent on the network name of the host on which it is being executed.

These scripts copy the preconfigured configuration files that do not change between dcache installations in their proper sub-paths in /opt/d-cache. The scripts also copy dcache templates into corresponding dcache config files, and then edit those files according to our needs for the configuration files that might change between dcache releases. Pool configuration rpm script verifies the existence of the proper pool directories and regenerate the pool setup files.

The configuration scripts remove the batch files for dcache services that are not supposed to run on the specific nodes. After that they execute dCache's /opt/d-cache/install/install.sh script that generates the startup scripts for each of the batch file left, and runs the srm deployment process on srm node.

After that dcache system is supposed to be fully configured and ready for production.

Dependencies

Dcache system operations depends on the presence of a number of programs and services that are not part of dcache server rpm:

Bash

Sun (Oracle) Java virtual machine version 1.6 or later.

Postgres database used by PnfsManager, Pnfs Companion, cleaner, billing, PinManager and SRM

Pnfs namespace

python

Enstore encp client and real-encp.sh scripts used by pools.

Dcache-fermi-config product checked out from cvs in ~enstore/dcache-deploy/dcache-fermi-config

dcache-fermi-config/kdcmux/kdcmux.py for kerberos door nodes (fndca[34])

Preparation work

Dcache-admin-stkend and dcache-pool-stken configuration rpms

Sources of the dcache configuration rpms are located in dcache-fermi-config cvs repository. They were located in cdftst/etc directory and used to capture the logic that applied to cdf, cms and pubic dcache installations and were based on dCache server 1.7.

When public dCache was updated to 1.9.2 last Summer, the scripts were updated, but the changes were kept locally to preserve the scripts compatibility with 1.7 dCache still in use in production.

I have collected the changes that were made for dCache 1.9.2 dcache installation and comitted the updated scripts to the new area stken/rpm. I removed logic that does not apply to public dCache to simplify the sripts and worked on making the sripts correctly configure the new test dcache installed on dmsdca cluster. Many of the services were now fully configured with the old configuration rpm, and I worked on making sure that all the changes that we had to make manually after the upgrade are now performed automatically.

For testing of the correctness of the configuration of the new system, I used pageDcache product, which performs a write and a read though each of the doors configured in a dCache. For dmsdca cluster I had to create a new configuration file that lists the hosts and ports of doors of each type.

After I got to the point when dCache 1.9.2 installed in conjunction with our configuration rpms would work out of the box, without manual modification whatsoever, I tagged the configuration rpm sources and worked on updating the configuration rpms to work with dCache server 1.9.5

real-encp.sh

Enstore project now made an rpm based version of the encp available. In order to take advantage of this, Dmitry worked on the modification of the real-encp.sh, so that it can work with encp that is installed using both ups/upd and rpm. He also made real-encp not depend on the enstore sources.

pnfs

Versions of the pnfs product available from dcache.org do not support postgres / os combination that we run on stkensrv1n. Vijay worked on building a pnfs rpm from sources on dmsdca08 with the same versions of SL and postgres as used in production.

Testing

Dcache 1.9.5 software is running in the majority of the dCache sites worldwide and is running in US-CMS T1 production. So the core and grid functionality of Dcache can be considered well tested. But there is a number of features that are unique to fermilab public dCache and CDF that have not been tested or used in production. These are kerberos and weak FTP doors and kerberos dcap doors. Special attention was payed to the deployment, configuration and operations of these components.

Testing was performed on the new dmsdca developer's test cluster.

Test system configuration

Here is the list of test system nodes and services running on these nodes
dmsdca03 (equivalent of fndca3a): lm, dcache (PoolManager), admin, billing, httpd, kebrosftpdoor
dmsdca05 (equivalent of fndca3a): dcap doors 00,01, 02 and 03, keberos dcap doors 00 and 01, gsi dcap doors 00 and 01, kerberos ftp door, gsi ftp doors GFTP0 and GFTP1, weak ftp door, SRM.
dmsdca08 (equivalent of stkensrv1n): pnfs, pnfs manager, cleaner, pnfs restored from latest public dCache pnfs backup
dmsdca01 and dmsdca02 (pools) : encp, real-encp.sh, pool services.

Testing steps

1. dmsdcaXX nodes configured with versions of OS, java and other products to match configuration in production, see addendum for a more detailed list of what was installed on dmsdcaXX to make it work
2. production pnfs rpm installed on dmsdca08, pnfs is restored from backup
3. dcache-server-1.9.2-9 rpm (current prodution) rpm installed on dmsdca01, dmsdca02, dmsdca03, dmsdca05, dmsdca08
4. dcache-pool-stken-1.9.2-9 rpm installed on dmsdca01,dmsdca02
5. dcache-admin-stken-1.9.2-9 rpm installed on dmsdca03,dmsdca05 and dmsdca08
6. files are written and observed to go to enstore, deleted from dcache, and read back though dcache, observed to be read from enstore and served to the client.
7. PageDcache script run, which tested all the doors present in the system.

8. Dcache-server-1.9.5-12 (future prodution) rpm installed on dmsdca01, dmsdca02, dmsdca03, dmsdca05, dmsdca08
9. dcache-pool-stken-1.9.5-12 rpm installed on dmsdca01,dmsdca02
10. dcache-admin-stken-1.9.5-12 rpm installed on dmsdca03,dmsdca05 and dmsdca08
11. all read write tests (6, 7 above) are repeated
12. Pnfs server upgrade to future production, all read write tests (6, 7 above) are repeated
13. new encp and real-encp.sh are installed on dmsdca01 and dmsdca02, all read write tests (6, 7 above) are repeated
14. Deleted the file from pnfs and make sure that the file is deleted from the pool
15. Run shutdown, verify everything is stopped, run startup, run tests again.

Still to do:

4. Ask Database Group (RMAN), OSG/Fermigrid and Minos to write and read files using their tools and procedures.
5. Test dCache pool upgrade procedures (developed in the documentation step) by a designated representative of the SSA group

Documentation

Document all the steps made during the test dCache upgrade

Prepare checklists for things to verify before during and after the upgrade

Document pool upgrade procedure, so that a member of SSA group can perform upgrade of the pool without dCache developer's involvement. All

Precious files draining prior to upgrade

We should ask users not to write into the system 12 hours before the upgrade .
We will change the configuration in PoolManager so that writes will not be possible.
Main goal is to make sure that all precious files are written to enstore. We will make sure that all precious files that are deleted from pnfs are removed from the pools. Further we will make sure that all files that are already written to enstore but are still marked as precious are changed to “Cached” State.

We will also distribute the dCache rpm and dCache admin/pool configuration rpms on the dCache nodes

Preparation, Shutdown and Upgrade

Preparation, March 03, 2010

- Put dcache server 1.9.5-16 rpm to /opt area of dcache pool and admin nodes
- Put dcache-admin-stken 1.9.5-16 rpm on dcache admin nodes
- Put dcache-pool-stken 1.9.5-16 rpm on dcache pool nodes
- Do the backup of the dcache area using pre-upgrade backup admin nodes part of https://plone4.fnal.gov/P0/DCache/admin/fndca-upgrade-plans-and-procedures/fndca-pool-pre-ugrade-backup-and-post-upgrade-rollback/
- Change PoolManager configuration so that the writes are not possible in the write pools configured to work with enstore. This should be done at 05 AM of March 04

  In admin interface the following commands will remove the write links, which will prevent users from writting into dCache:

  ```
  [enstore@fndca2a ~]$ adcache

      dCache Admin (VII) (user=enstore)


  [fndca3a.fnal.gov] (local) enstore > cd PoolManager
  [fndca3a.fnal.gov] (PoolManager) enstore > psu remove link write-link
  [fndca3a.fnal.gov] (PoolManager) enstore > psu remove link RawDataWrite-link
  [fndca3a.fnal.gov] (PoolManager) enstore > psu remove link KteVWrite-link
  [fndca3a.fnal.gov] (PoolManager) enstore > psu remove link DES-link
  fndca3a.fnal.gov] (PoolManager) enstore > ..
  [fndca3a.fnal.gov] (local) enstore > logoff
  dmg.util.CommandExitException: (0) Done
  [fndca3a.fnal.gov] (local) enstore > Connection to fndca3a.fnal.gov closed.
  [enstore@fndca2a ~]$
  ```

- Pnfs rpm is copied to stkensrv1n (pnfs node)

Shutdown, March 04, 2010, 8AM

- dCache is shut down using cold-stop.new script
- Pnfs is shut down

Upgrade.

- Starting at 8:30 AM on March 04, 2010
- https://plone4.fnal.gov/P0/DCache/admin/fndca-upgrade-plans-and-procedures/fndca-pool-pre-ugrade-backup-and-post-upgrade-rollback/ contains description of the backup steps of the previously installed dcache and steps that could be taken to roll back the upgrade in case of failure
- On pool nodes all the control directories with control files and pool setup files are backed up into /diskX/pool-dir/control-keep-Feb-18 directories.

- Dcache server rpms are installed on all nodes
- Dcache-admin-stken rpms are installed on admin nodes
- https://plone4.fnal.gov/P0/DCache/admin/fndca-upgrade-plans-and-procedures/fndca-pool-installation/
-
- Dcache-pool-rpms are installed on pool nodes (see Appendix 1 for details)
- pnfs is upgraded on stkensrv1n (see Appendix 2 for details)
- Dcache admin pnfs node services are brought up
- dCache is brought up
- dCache functionality and monitoring is verified
- Each pool as verified as being online
- All central dCache services are verified as being online
- All doors are verified to function properly by Running Page Dcache on fndca2a
- System released for the limited testing
- System is released to the users at 2PM.
- In case of complications we leave the right to extend the downtime till the end of the workday

Contingency plan

Since we backed up the whole dCache configuration and control directories, moving these back instead of the newly created ones should allow us to go back to previous 1.9.2 functionality in no time at all. If the new files are written we will make sure that they are flashed to tape before rolling back to the original configuration.

People

SSA – dCache shutdown according to their procedures
Timur – preparation/testing/documentation/dCache head nodes upgrade, pnfs upgrade
SSA - dCache pools upgrades.

Appendix 1, Installing encp on pools

By Dmitry Litvintsev

```
yum --enablerepo=enstore install encp-ups-opt-dcache.i386
yum --enablerepo=enstore install encp-conf-FNAL.noarch
cd ~enstore/dcache-deploy/dcache-fermi-config/scripts
cvs update -A real-encp.sh
```

Apendix 2, Upgrading Pnfs Server on production System

By Vijay Sekhri

Prerequisite

Before we do PNFS upgrade we need to stop all dcache services running on our production nodes and unmount pnfs.

1) **Log into the host running pnfs server as root**. For our production system it is stkensrv1n.fnal.gov

```
ssh root@stkensrv1n.fnal.gov
```

2) **Get the correct RPM for PNFS server**. The following RPM is built for SL5. Get another one for SL4.4. Notice that the following RPM will not work on our production system because it is SL4.4

```
cd /opt/RPMS
wget http://www.desy.de/~paul/tmp/pnfs-postgresql-3.1.18-1.x86_64.rpm
```

There is RPM especially built for our production system available at dmsdca08.fnal.gov@/usr/src/redhat/RPMS/x86_64/pnfs-postgresql-3.1.18-1.x86_64.rpm
Get it from there.

3) **Stop any dcache services running on this system**. PnfsManager should already be running on this system. Check in the stdout to see if pnfsDomain was indeed stoped

```
/opt/d-cache/bin/dcache stop
Stopping dirDomain (pid=29919) 0 Done
Stopping pnfsDomain (pid=29754) 0 1 Done
```

4) **Grep for java processes** to make sure none of the dcache services are running.

```
ps -aef | grep java
root 26492 18601 0 10:57 pts/0 00:00:00 grep java
```

5) **Grep for pnfs mount** on this system and you should see a localhost mount like below

```
mount | grep pnfs
localhost:/fs on /pnfs/fs type nfs
(rw,noatime,nodiratime,udp,intr,noac,hard,nfsvers=2,addr=127.0.0.1)
```

6) **Grep for pnfs server processes** and you should see a bunch of processes like below

```
ps -aef | grep pnfsd
root 1425 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1426 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1427 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1428 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1429 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1430 1 0 Jan26 ? 00:00:32 ./pnfsd
root 1431 1 0 Jan26 ? 00:00:33 ./pnfsd
root 1433 1 0 Jan26 ? 00:00:32 ./pnfsd
root 1434 1 0 Jan26 ? 00:00:34 ./pnfsd
root 1435 1 0 Jan26 ? 00:00:33 ./pnfsd
root 25245 18601 0 10:42 pts/0 00:00:00 grep pnfs
```

7) **Stop the pnfs server** with the following command. Notice that it will stop the pnfs

server and unmount pnfs as well.

```
/etc/init.d/pnfs stop
Shutting down pnfs services (PostgreSQL version):
Stopping Heartbeat .... Ready
Killing pnfsd Done
Killing pmountd Done
Killing dbserver . Done
Removing 12 Clients 0+ 1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+
Removing 70 Servers 0+ 1+ 2+ 3+ 4+ 5+ 6+ 7+ 8+ 9+ 10+ 11+ 12+ 13+ 14+ 15+ 16+ 17+
18+ 19+ 20+ 21+ 22+ 23+ 24+ 25+ 26+ 27+ 28+ 29+ 30+ 31+ 32+ 33+ 34+ 35+ 36+ 37+
38+ 39+ 40+ 41+ 42+ 43+ 44+ 45+ 46+ 47+ 48+ 49+ 50+ 51+ 52+ 53+ 54+ 55+ 56+ 57+
58+ 59+ 60+ 61+ 62+ 63+ 64+ 65+ 66+ 67+ 68+ 69+
Removing main switchboard ... O.K.
```

8) **Grep for pnfs server** process again and now you should not see any process related to pnfs

```
ps -aef | grep pnfsd
root 25541 18601 0 10:43 pts/0 00:00:00 grep pnfs
```

Notice we do have prostgres running on this system and it does not need to be stopped. If you see any pnfs process still running, try killing it by hand. Pnfs processes should all stop before we do the upgrade.

9) **Grep for pnfs mount** and this time you should not see localhost mounted

```
mount | grep pnfs
```

If you still see any mount , then try removing forcefully

```
umount --force /pnfs
```

10) **Save the pnfs startup script** because it will get overwritten by the RPM

```
cp /etc/init.d/pnfs /etc/init.d/pnfs.save
```

11) **Upgrade the pnfs server** with following command

```
rpm -Uvh pnfs-postgresql-3.1.18-1.x86_64.rpm
Preparing... ########################################### [100%]
1:pnfs-postgresql ########################################### [100%]
```

12) **Restore the original pnfs startup script**

```
cp /etc/init.d/pnfs.save /etc/init.d/pnfs
```

13) **Start the pnfs server** with following command and look at the stdout carefully

```
/etc/init.d/pnfs start
Starting pnfs services (PostgreSQL version):
Shmcom : Installed 12 Clients and 70 Servers
Starting database server for admin (/srv1/pnfs/db/admin) ... O.K.
Starting database server for eagle (/srv1/pnfs/db/eagle) ... O.K.
Starting database server for NULL (/srv1/pnfs/db/NULL) ... O.K.
Starting database server for theory (/srv1/pnfs/db/theory) ... O.K.
Starting database server for NeST (/srv1/pnfs/db/NeST) ... O.K.
Starting database server for test (/srv1/pnfs/db/test) ... O.K.
Starting database server for beamstool (/srv1/pnfs/db/beamstool) ... O.K.
Starting database server for sdss (/srv1/pnfs/db/sdss) ... O.K.
Starting database server for lqcd (/srv1/pnfs/db/lqcd) ... O.K.
Starting database server for sdssapo3 (/srv1/pnfs/db/sdssapo3) ... O.K.
Starting database server for cepa (/srv1/pnfs/db/cepa) ... O.K.
Starting database server for selex (/srv1/pnfs/db/selex) ... O.K.
Starting database server for cdms (/srv1/pnfs/db/cdms) ... O.K.
Starting database server for cdf (/srv1/pnfs/db/cdf) ... O.K.
Starting database server for BDMS (/srv1/pnfs/db/BDMS) ... O.K.
Starting database server for minos (/srv1/pnfs/db/minos) ... O.K.
Starting database server for blastman (/srv1/pnfs/db/blastman) ... O.K.
Starting database server for d0lib-archive (/srv1/pnfs/db/d0lib-archive) ... O.K.
Starting database server for netflow (/srv1/pnfs/db/netflow) ... O.K.
Starting database server for auger (/srv1/pnfs/db/auger) ... O.K.
Starting database server for ccf (/srv1/pnfs/db/ccf) ... O.K.
Starting database server for beagle (/srv1/pnfs/db/beagle) ... O.K.
Starting database server for des (/srv1/pnfs/db/des) ... O.K.
Starting database server for btev (/srv1/pnfs/db/btev) ... O.K.
Starting database server for p929 (/srv1/pnfs/db/p929) ... O.K.
Starting database server for ckm (/srv1/pnfs/db/ckm) ... O.K.
Starting database server for sdssapo4 (/srv1/pnfs/db/sdssapo4) ... O.K.
Starting database server for e831 (/srv1/pnfs/db/e831) ... O.K.
Starting database server for sdssapo5 (/srv1/pnfs/db/sdssapo5) ... O.K.
Starting database server for miniboone (/srv1/pnfs/db/miniboone) ... O.K.
Starting database server for sdssdb2 (/srv1/pnfs/db/sdssdb2) ... O.K.
Starting database server for ktev (/srv1/pnfs/db/ktev) ... O.K.
Starting database server for sdsstam0 (/srv1/pnfs/db/sdsstam0) ... O.K.
Starting database server for E815 (/srv1/pnfs/db/E815) ... O.K.
Starting database server for sdssapo0 (/srv1/pnfs/db/sdssapo0) ... O.K.
Starting database server for E872 (/srv1/pnfs/db/E872) ... O.K.
Starting database server for sdssapo1 (/srv1/pnfs/db/sdssapo1) ... O.K.
Starting database server for e791 (/srv1/pnfs/db/e791) ... O.K.
Starting database server for sdssapo2 (/srv1/pnfs/db/sdssapo2) ... O.K.
Starting database server for e740 (/srv1/pnfs/db/e740) ... O.K.
Starting database server for Migration (/srv1/pnfs/db/Migration) ... O.K.
Starting database server for patriot (/srv1/pnfs/db/patriot) ... O.K.
Starting database server for e871 (/srv1/pnfs/db/e871) ... O.K.
Starting database server for e907 (/srv1/pnfs/db/e907) ... O.K.
Starting database server for e835 (/srv1/pnfs/db/e835) ... O.K.
Starting database server for cdf-sam (/srv1/pnfs/db/cdf-sam) ... O.K.
Starting database server for sdssImaging (/srv1/pnfs/db/sdssImaging) ... O.K.
Starting database server for sdssSpectro (/srv1/pnfs/db/sdssSpectro) ... O.K.
Starting database server for exp-db (/srv1/pnfs/db/exp-db) ... O.K.
Starting database server for coupp (/srv1/pnfs/db/coupp) ... O.K.
Starting database server for astro (/srv1/pnfs/db/astro) ... O.K.
Starting database server for minosReco1 (/srv1/pnfs/db/minosReco1) ... O.K.
Starting database server for booneTD1 (/srv1/pnfs/db/booneTD1) ... O.K.
Starting database server for fermigrid (/srv1/pnfs/db/fermigrid) ... O.K.
Starting database server for cstlogs (/srv1/pnfs/db/cstlogs) ... O.K.
Starting database server for minerva (/srv1/pnfs/db/minerva) ... O.K.
Starting database server for sciboone (/srv1/pnfs/db/sciboone) ... O.K.
Starting database server for ilc_stop (/srv1/pnfs/db/ilc_stop) ... O.K.
Starting database server for ilc4c (/srv1/pnfs/db/ilc4c) ... O.K.
Starting database server for das (/srv1/pnfs/db/das) ... O.K.
Starting database server for ssa_test (/srv1/pnfs/db/ssa_test) ... O.K.
Starting database server for ilc (/srv1/pnfs/db/ilc) ... O.K.
Starting database server for mu2e (/srv1/pnfs/db/mu2e) ... O.K.
Starting database server for argoneut (/srv1/pnfs/db/argoneut) ... O.K.
Starting database server for nova (/srv1/pnfs/db/nova) ... O.K.
Waiting for dbservers to register ... Ready
Starting Mountd : pmountd
Starting nfsd : pnfsd
```

14) **Grep for pnfs server** process once again and now you should see a whole bunch of processes running

```
ps -aef | grep pnfsd
root 2770 21730 0 15:48 pts/0 00:00:00 grep pnfsd
root 31537 1 0 15:00 ? 00:00:00 ./pnfsd
root 31538 1 0 15:00 ? 00:00:00 ./pnfsd
root 31539 1 0 15:00 ? 00:00:00 ./pnfsd
root 31540 1 0 15:00 ? 00:00:00 ./pnfsd
root 31541 1 0 15:00 ? 00:00:00 ./pnfsd
root 31542 1 0 15:00 ? 00:00:00 ./pnfsd
root 31543 1 0 15:00 ? 00:00:00 ./pnfsd
root 31545 1 0 15:00 ? 00:00:00 ./pnfsd
root 31546 1 0 15:00 ? 00:00:00 ./pnfsd
root 31547 1 0 15:00 ? 00:00:00 ./pnfsd
```

If you do not see any processes then the pnfs server did not start properly. Look in /srv1/pnfs/log/pnfsd.log file to see cause of errors abd try to fix them. The log file location is mentioned in /usr/etc/pnfsSetup

15) **Grep for pnfs mount** on this system and now you should see a localhost mount like below

```
mount | grep pnfs
localhost:/fs on /pnfs/fs type nfs
(rw,noatime,nodiratime,udp,intr,noac,hard,nfsvers=2,addr=127.0.0.1)
```

If you do not see the mount, again check in /srv1/pnfs/log/pnfsd.log to determine the cause of the problem. It is imperative that the /etc/init.d/pnfs start properly and mounts the pnfs server locally. If you fix it by hand then when the system restarts , you will again get into the same problem. So make sure that pnfs is started properly with the init script

16) **Start your dcache services** that should include PnfsManager with following command

```
/opt/d-cache/bin/dcache start
Fri Jan 29 15:49:27 CST 2010 Checking on /tmp
Fri Jan 29 15:49:27 CST 2010 Checking if directories that we need exist
Fri Jan 29 15:49:27 CST 2010 Checking if pnfs is mounted
Starting pnfsDomain 6 Done (pid=3038)
Fri Jan 29 15:49:28 CST 2010 Checking on /tmp
Fri Jan 29 15:49:28 CST 2010 Checking if directories that we need exist
Fri Jan 29 15:49:28 CST 2010 Checking if pnfs is mounted
Starting dirDomain 6 Done (pid=3210)
```